

CString(Format)

Be careful with string formatting operations

Sean Barnum, Digital, Inc. [vita¹]

Copyright © 2007 Digital, Inc.

2007-03-22

Part "Original Digital Coding Rule in XML"

Mime-type: text/xml, size: 4256 bytes

Attack Category	• Malicious Input											
Vulnerability Category	• Format string											
Software Context	• String Formatting											
Location	• Cstring (MFC)											
Description	<p>CString formatting methods are vulnerable to format string attacks.</p> <p>The CString class represents a string of ANSI or Unicode characters. It includes member functions for formatting that have functionality much like that of sprintf. These methods are vulnerable to format string attacks.</p>											
APIs	<table border="1"><thead><tr><th>Function Name</th><th>Comments</th></tr></thead><tbody><tr><td>CString::Format</td><td>Src: 1 variable; Fmt: 0; No dest arg, dummy, val</td></tr><tr><td>CString::FormatMessage</td><td>Src: 1 variable; Fmt: 0; No dest arg, dummy, val</td></tr><tr><td>CString::FormatMessageV</td><td>Src: 1 variable; Fmt: 0; No dest arg, dummy, val</td></tr><tr><td>CString::FormatV</td><td>Src: 1 variable; Fmt: 0; No dest arg, dummy, val</td></tr></tbody></table>		Function Name	Comments	CString::Format	Src: 1 variable; Fmt: 0; No dest arg, dummy, val	CString::FormatMessage	Src: 1 variable; Fmt: 0; No dest arg, dummy, val	CString::FormatMessageV	Src: 1 variable; Fmt: 0; No dest arg, dummy, val	CString::FormatV	Src: 1 variable; Fmt: 0; No dest arg, dummy, val
Function Name	Comments											
CString::Format	Src: 1 variable; Fmt: 0; No dest arg, dummy, val											
CString::FormatMessage	Src: 1 variable; Fmt: 0; No dest arg, dummy, val											
CString::FormatMessageV	Src: 1 variable; Fmt: 0; No dest arg, dummy, val											
CString::FormatV	Src: 1 variable; Fmt: 0; No dest arg, dummy, val											
Method of Attack	<p>If an attacker can control formatting text, a Format string attack could be mounted.</p> <p>These functions can be abused in much the same way sprintf() can be. If the attacker embeds a series of "%s" format fields in the data, the function will interpret them as formatting commands. It will get the corresponding data values off the stack. With enough "%s" fields, the function will eventually dereference a bad pointer (e.g. NULL) and crash. If the attacker includes "%n" fields, the function will write values to memory, which may ultimately be exploitable.</p>											
Exception Criteria												

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

Solutions	Solution Applicability	Solution Description	Solution Efficacy				
	Generally applicable to CString format methods.	Do not use a format string that may come from or have been tampered with by an untrustworthy source.	Effective.				
Signature Details	<pre>void __cdecl CString::Format(UINT nFormatID, [, argument]...); void __cdecl CString::Format(PCXSTR pszFormat, [, argument]...); void __cdecl CString::FormatMessage(UINT nFormatID, [, argument]...); void __cdecl CString::FormatMessage(PCXSTR pszFormat, [, argument]...); void CString::FormatV(PCXSTR pszFormat, va_list args);</pre>						
Examples of Incorrect Code	<pre>CString aString; aString.Format(userSuppliedText);</pre>						
Examples of Corrected Code	<pre>CString aString; aString.Format("%s" , userSuppliedText);</pre>						
Source Reference	http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vclib/html/vclrfcstringformatmessage.asp ²						
Recommended Resource	MSDN reference for CString ³						
Discriminant Set	<table border="1"> <tr> <td>Operating System</td><td>• Windows</td></tr> <tr> <td>Language</td><td>• C++</td></tr> </table>			Operating System	• Windows	Language	• C++
Operating System	• Windows						
Language	• C++						

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Digital, including information about “Fair Use,” contact Digital at copyright@digital.com¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@digital.com>